

TARTU ÜLIKOOL
MATEMAATIKAINFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Infotehnoloogia eriala

Rauno Kiss

**Intelligentne assistent
argumenteerimise harjutamiseks**

Bakalaureusetöö (6 AP)

Juhendaja: prof. M.Koit

Autor: “.....“ juuni 2011
Juhendaja: “.....“ juuni 2011

Lubada kaitsmisele

Professor “.....“ juuni 2011

TARTU 2011

Sisukord

Sissejuhatus	3
1. Dialoogiteooria	4
1.1 Dialoogi teooriad ja mudelid	4
1.2 Tähelepanekud dialoogi modelleerimisel	5
2. Programmi tutvustus	6
2.1 Kasutajaliides	6
2.2 Algoritm	8
2.3 Dialoogide salvestamine	8
3. Kasutusjuhend	9
3.1 Keskkonna ülesseadmine	9
3.2 Dialoogi alustamine	10
3.3 Vestluse salvestamine	11
4. Edasiarendamise võimalused	12
5. Mõned teised süsteemid	13
6. Kokkuvõte	14
7. Intelligent assistant for practicing argumentation	15
8. Kasutatud kirjandus	16
9. Lisad	17
Lisa 1: Teema valimise leht (kaart kaks)	17
Lisa 2: Väljavõte dialoogi logifailist.	17
Lisa 3. Dialoogi salvestamise ekraanipilt	18
Lisa 4. Algoritmi plokk-skeemid	19

Sissejuhatus

Töö eesmärgiks on luua tehisintellekti realiseeriv süsteem, mis aitaks kasutajal harjutada kirjalikku argumenteerimist eesti keeles. Programm peaks olema kasutatav lokaalselt käivitades Java rakendusena, mis tagab aplikatsiooni töötamise ka interneti puudumisel. Kuna töö tegemise hetkel polnud võimalik avalikult kätte saada samasuguseid tehisintellekte, siis tuli rakenduse loomist alustada nullist. Aluseks on võetud 2003. aastal ilmunud artikkel „Eestikeelse dialoogi modelleerimine“ [1], kust sai vajalikku informatsiooni dialoogi modelleerimise kohta. Selles artiklis on välja toodud ka mõningaid näiteid algoritmide väljatöötamisest, kust oli võimalik saada ideid rakenduse loomisel.

Uurimustöö käigus valminud rakendus on alustalaks suuremale projektile. Antud rakendus loob lihtsa tehisintellekti süsteemi, mis „vaidleb“ kasutajaga, kasutades etteantud argumente. Käesoleva projekti edasiarendus tulevikus oleks rakenduse viimine kõrgemale tasemele, mis eeldaks eesti keele sõnade semantilise, morfoloogilise ja süntaktilise analüüsi sissetoomist.

Bakalaureusetöö ülesehituse võib jagada kolmeks osaks – rakenduse alus, kirjeldus ja analüüs. Esimeses osas antakse ülevaade dialoogi teooriast. Tuuakse erinevaid mudeleid koos kirjelduste ja võrdlustega. Samuti esitatakse tähtsamad tähelepanekud dialoogi modelleerimisel, mida rakenduse loomise juures tuli arvestada. Teine osa kirjeldab programmi struktuuri, ülesehitust ja kujundust. Lisaks antakse ülevaade rakenduse loomisel kasutatud tehnoloogiatest, lisapakettidest ja algoritmidest. Samuti leidub selles peatükis rakenduse paigaldamise ja kasutamise juhend. Kolmandas osas analüüsitakse bakalaureusetöös valminud rakenduse kasutusala, tuuakse võrdlusi olemasolevate rakendustega ja kavandatakse edasiarendamise võimalusi.

Töö lisades on toodud:

- dialoogi teema valimise ekraanipilt,
- logifaili näidis,
- dialoogi salvestamise ekraanipildid,
- algoritmi seletavad plokk-skeemid.

Bakalaureusetöoga on kaasas DVD, millel leiduvad:

- tarkvara lähtekood,
- rakenduse käivitamise failid,
- kasutusjuhend,
- bakalaureusetöö kirjalik osa.

1. Dialoogiteooria

Uurimustööga kaasnes dialoogi teooria ja mudelite uurimine, mis on oluline dialoogi modelleerimise juures. Sõnal *dialoog* on Eesti keele seletava sõnaraamatu [2] järgi mitu tähendust. Näiteks tähendab see kahekõnet, kõnelust kahe või mitme isiku vahel, vestlust, mõttevahetust. Arvestades neid tähendusi ja järgides töö skoopi, saab sõna *dialoog* vaadelda kahe nurga alt: 1) dialoog kui suhtlusprotsess, milles on kaks või enam osalejat, kes vahetavad omavahel lauseid, väiteid, argumente, ja 2) dialoogi saab vaadelda ka kui ajas kulgevat protsessi, mille keskmeks on seotud tekst või kõne. Sellest tulenevalt tähistab mõiste *dialoogiteooria* teooriat, mis uurib kahe osapoolte vahel toimuvaid vestlusi, teostab analüüsi ja loob erinevaid mudeleid. Tänapäeval tuntakse kolme erinevat dialoogiteooriat [1]: dialoogigrammatika, plaanipõhine ja ühistegevuse teooria. Need teooriad täiendavad üksteist ning koos aitavad kirjeldada või luua võimalikult loomulikku vestlust dialoogi osapoolte vahel.

1.1 Dialoogi teooriad ja mudelid

Dialoogigrammatika järgi esineb dialoogis teatud arv järjestikuseid regulaarsusi. Näiteks küsimusele järgneb vastus, ettepanekule selle vastuvõtmine või tagasilükkamine jne. Antud teooria käsitleb dialoogi kui selliste sündmuste järjendit. Samuti eristatakse dialoogi osi nagu näiteks küsimus, vastus, ettepanek, tagasilükkamine jne. Teooria järgi on dialoogil ka erinevad staadiumid nagu initsieerimine, reageerimine ja hindamine.

Plaanipõhise dialoogiteooria järgi väljendavad dialoogis osalejate laused suhtlustegevusi ehk kõneakte. Inimesed, kes omavahel suhtlevad, teevad ettepanekuid, informeerivad ja hoiatavad üksteist jne. Osalejad planeerivad oma tegevusi ette, eesmärgiga saavutada eelnevalt kehtestatud siht või vähemalt muuta kaaslaste hoiakut. Lisaks võimaldab see teooria lahendada ka nn kaudse kõneakti probleemi. Näiteks kui dialoogi üks osapool lausub: „Telekapult peaks olema kusagil diivani peal“, siis teine dialoogis osaleja peab uskuma, et telekapult tõesti eksisteerib, aga lisaks peab ta ka aru saama, et ütleja plaaniks oli küsida või panna ta otsima seda objekti.

Ühistegevuse teooria järgi on dialoog protsess, mida osalejad viivad läbi üheskoos. Kõik dialoogis osalejad kannavad vastutust dialoogi kestmise eest. See teooria sarnaneb plaanipõhise dialoogiteooriaga, kuid annab lisaks veel mõningaid aspekte. Ühistegevuse teooria toob vestlusesse sisse sotsiaalse aspekti. See tähendab, et mõlemad osapooled peavad töötama eesmärgi või plaani saavutamise nimel. Selline tegevus tagatakse muuhulgas üleküsimiste, selgituste, kinnituste küsimisega partnerilt. Rõhk on pandud koostööle ja tagasisidele.

1.2 Tähelepanekud dialoogi modelleerimisel

Dialoogi modelleerimist defineeritakse kui suhtlusprotsessi formaalset kirjeldamist. Arvutiga toimuva dialoogi modelleerimise juures tuleks võimaldada inimese suhtlus arvutiga inimesele loomulikus keeles. Lisaks tuleb arvestada inimvahelise suhtluse kirjutamata reeglitega. Tulenevalt sellest ongi loogiline, et dialoogis küsimusele järgneb vastus ja ettepanekule järgneb selle vastuvõtmine või tagasilükkamine. Sellise dialoogi modelleerimine nõuab algoritmide väljatöötamist, mis võimaldaksid arvutil pidada loomulikku vestlust inimesega.

Inimese ja arvuti vahel toimuva loomuliku vestluse aitavad tagada suhtlusstrateegiad ja –taktikad. Siinkohal tekib seos strateegia ja plaanipõhise dialoogiteooria vahel. Vastavalt plaanipõhisele dialoogiteooriale, seatakse dialoogi eesmärgiks teatav plaan, mida soovitakse partnerit realiseerima panna. Eesmärgi saavutamine sõltub kasutajale lähenemise strateegiast. Levinuimateks suhtlusstrateegiateks on näiteks ahvatlemine, veenmine, ähvardamine. Ahvatlemise juures üritab agent (arvuti) suurendada partneri meeldivust eesmärgi saavutamise suunas. Veenmise puhul viitab agent eesmärgi kasulikkusele ning ähvardamise puhul rõhub agent tegevuse tegemata jätmisele järgnevale karistusele.

Rakenduse ehitamisel kasutati käesolevas töös kahe dialoogiteooria abi algoritmi väljatöötamisel, mis suudaks argumenteerida kasutajaga. Tunnusjooni võeti nii dialoogigrammatikast kui ka plaanipõhisest teooriast. Dialoogigrammatikaga määratakse see, et rakenduses saab vestlus toimuda ainult kordamööda – üks dialoogi osapool esitab küsimuse või argumendi ning teine osapool esitab vastuargumendi või nõustub esimesega. Argumentide loomisel kasutati suhtlusstrateegiate ideed. Argumendid jagati nelja suuremasse rühma – veenmine (meeldivust suurendavad ja vähendavad argumendid), ahvatlemine (kasulikkust suurendavad ja vähendavad argumendid), ähvardamine (karistust rõhutavad argumendid) ja ressursside puudumine.

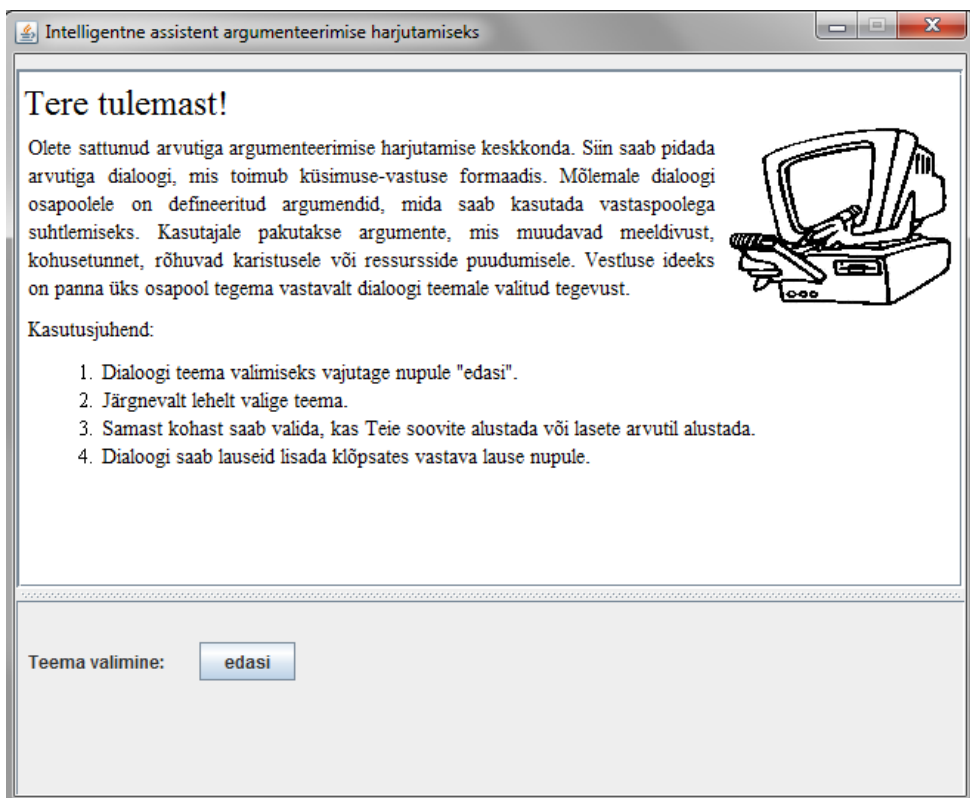
2. Programmi tutvustus

Rakenduse põhiideeks on visualiseerida dialoogi kasutajaga, kasutades etteantud argumente. Suhtlus seisneb selles, et vestluse alustaja A üritab teist osapoolt B panna tegema mingit tegevust D. Vestluse teema, ühtlasi ka tegevuse D, valib kasutaja kahe väljapakutud teema hulgast. Nendeks teemadeks on „Tööreis Jaapanisse“ ja „Lähme ülehommeme ujuma“. Samuti valib kasutaja, kes alustab dialoogi, kas arvuti või tema ise.

Tarkvara valmis Java iseseisva rakendusena. Programmeerimiskeeleks valiti Java, sest tänu oma lihtsusele rahuldab ta kõik antud uurimustöö vajadused ja nõuded. Java üheks plussiks on lihtsasti ülesseatavad graafilised objektid, mis lihtsustasid kasutajaliidese ehitamist. Programmeerimis- ja testkeskkonnana kasutati vabavaralist tarkvara Eclipse. Aplikaatsioonil on graafiline kasutajaliides, kus avaekraanil tervitatakse kasutajat ja antakse edasised juhendid programmis orienteerumiseks.

2.1 Kasutajaliides

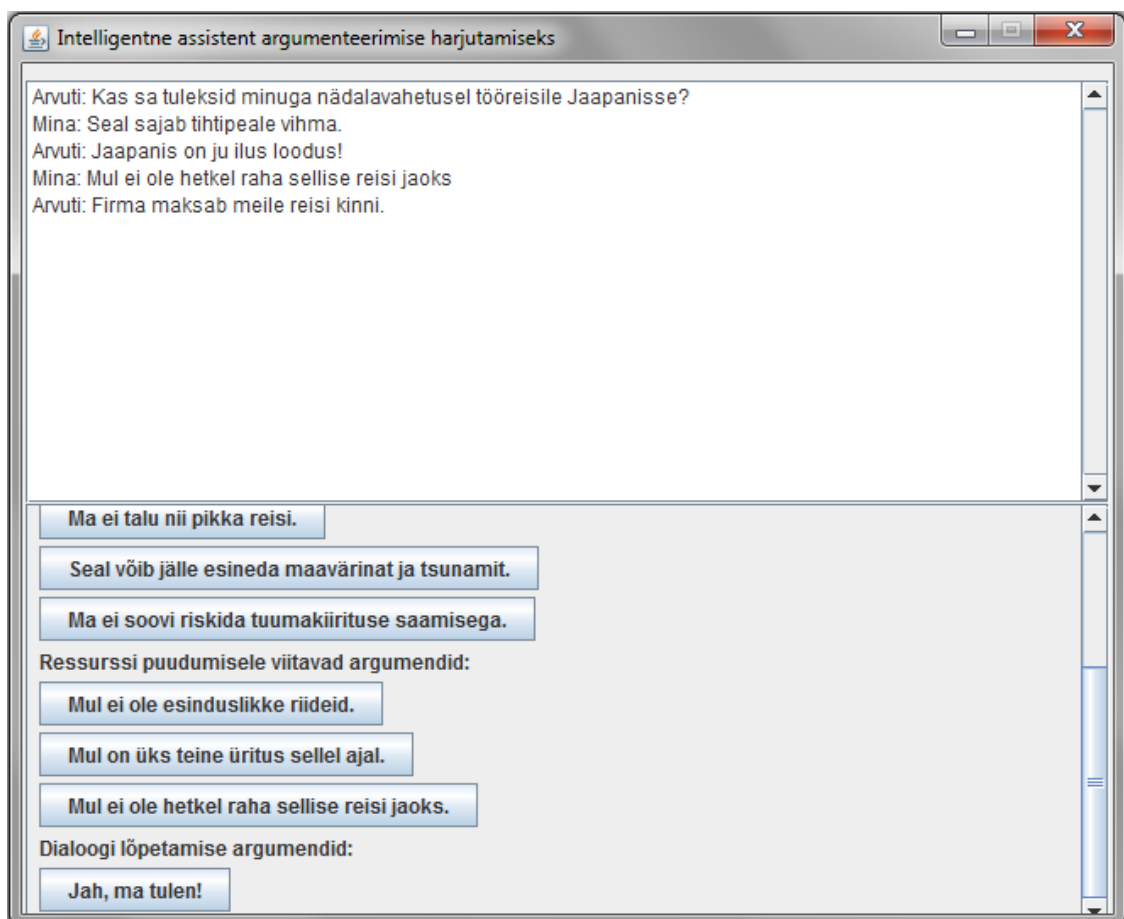
Rakenduse kasutajaliides on üles ehitatud, kasutades Java poolt pakutud malle (*layout*). Antud rakendus tugineb kaardi-malli (*cardlayout*) abil loodud kujundusele. Programmi käivitades luuakse 4 erinevat kaarti. Esimese kaardi peal luuakse avaleht, millel tervitatakse kasutajat, tutvustatakse lühidalt, mis programmiga on tegemist ja mida selle programmiga teha saab. Kasutajale antakse edasised juhtnöörid programmis orienteerumiseks.



Joonis 1. Rakenduse avaleht

Teisel kaardil luuakse teema valimise leht (vaata Lisa 1). Siin saab kasutaja valida dialoogi teema ja määrata vestluse alustaja. Kasutajal on praegu valida kaks teemat: „Tööreis Jaapanisse ja „Lähme ülehommene ujuma“. Teemade nimistu on loodud, kasutades objekti raadionupp. Selekteerides vastava teema ja klikkides vastavale nupule, kes alustab, edastatakse teema kaartidele kolm ja neli. Kaardid kolm ja neli uuendavad oma sisu vastavalt teiselt kaardilt saadud teemale.

Kolmandal ja neljandal kaardil luuakse kaks objekti: tekstiväli ja argumentide paneel. Teksti välja loomiseks kasutatakse Java poolt pakutud objekti *JTextArea*. Dialoogi argumentid kuvatakse *JButton* objektide näol lehe alumisse poolde objekti *JPanel* peale. Klikkides valitud argumenti peal, kuvatakse dialoogi lause teksti-välja sisse ning algoritm arvutab kaalude järgi, millist argumenti kasutajale järgmiseks vastu öelda. Sellest algoritmist tuleb täpsemalt juttu järgmises alampeatükis. Nuppude disainimisel ja kujundamisel kasutatakse Java lisateeki *MigLayout* [3]. *MigLayout* on kolmanda osapoole poolt arendatav vabavaraline ja avatud lähtekoodiga Java mall, mis lihtsustab objektide paigutamist Java graafilistes kasutajaliidestest, mis on ka ühtlasi selle lisateegi tugevaimaks küljeks. Ta kasutab string-tüüpi sõnesid, et vormindada kujundust.



Joonis 2. Dialoogi pidamise keskkond (kaart 4)

2.2 Algoritm

Rakenduses kasutatavad dialoogide argumendid jagati teema järgi klassidesse, milles moodustati argumentide liikide järgi sõnade massiivid. Aplikatsioonis on kasutusel kuut tüüpi argumente: meeldivust ja kasulikkust suurendavad argumendid, karistust rõhutavad argumendid, meeldivust ja kasulikkust vähendavad argumendid ning ressursi puudumisele viitavad argumendid. Igasse sõnade massiivi paigutati lihtsuse ja näitlikkuse mõttes 3 argumenti. Teema klassi failis leiduvad veel dialoogi alustamiseks ja lõpetamiseks vajalikud sõned.

Algoritmi väljatöötamiseks võeti üle mõningaid aspekte dialoogigrammatikast ja plaanipõhisest teooriast, nagu varem mainitud. Algoritm arvestab sellega, et dialoogis osalejad esinevad korda mööda ja dialoogi alustaja eesmärk on panna teine osaleja tegema (dialoogi teemaga määratud) tegevust. Lisaks on rakendatud artiklis [1] kirjeldatud kaalude süsteemi. Artiklist võeti osaliselt üle idee ja muudeti vastavalt rakenduse vajadusele. Igale argumentide liigile määrati oma kaal. Näiteks meeldivust suurendavatel argumentidel on kaal kolm, kasulikkust suurendavatel argumentidel viis ja karistust rõhutavate argumentide kaal on kümme. Algoritm peab arvestust selle üle, mida kasutaja vastab. Kui kasutaja valib mingi argumendi, siis jäetakse selle argumendi kaal meelde täisarvu-tüüpi vektorisse. Vastavalt kaalude summale, valib algoritm rakendatava strateegia. Näiteks kui kaalude summa on üle kümne, siis valib algoritm argumendi karistust rõhutavate argumentide hulgast. Sarnaselt: kui kaalude summa on viie ja kümne vahel, siis algoritm valib kasulikkuse rõhutamise strateegia ning kui kaalude summa on alla viie, siis algoritm rakendab meeldivuse suurendamise strateegiat. Dialoogi alustades on kaalude summa null. Selline kaalude süsteem tagab loomulikule vestlusele lähedase dialoogi, sest kaalud takistavad algoritmil näiteks kohe vestluse alguses hakata kasutajat ähvardama. Vaata ka lisa 4, kus on algoritmi tööd seletavad joonised.

2.3 Dialoogide salvestamine

Rakenduse põhiideeks oli võimaldada kasutajal harjutada argumenteerimist arvutiga. Sellest tulenevalt on loodud rakendusele lisavõimalus salvestada toimunud dialoogid. Vestluse käigus esinevaid lauseid hoitakse sõne-tüüpi vektoris. Kui vestluse käigus esitatakse dialoogi lõpetamise tingimus, s.t. teine osapool nõustub tegema teemale vastavat tegevust, siis esitatakse kasutajale uus aken. Selle aknaga teavitatakse kasutajat, et dialoog on edukalt lõppenud ning pakutakse kolme valikut: minna tagasi avalehele, salvestada dialoog või sulgeda programm. Valides dialoogi salvestamise, esitatakse kasutajale operatsioonisüsteemidest tuttav salvesta (*save*) aken, kasutades objekti *JFileChooser*. Kasutaja saab määrata dialoogi faili nime ning asukoha, kuhu fail salvestatakse. Logi-faili on võimalik salvestada ainult teksti-faili (.txt) formaadis (vaata lisa 2). Vajutades nuppu salvesta (*save*), viiakse kasutaja tagasi avalehele ning dialoogi salvestamine õnnestus. Lisas 3 on ka toodud mõningad pildid dialoogi salvestamise protsessist.

3. Kasutusjuhend

Kuna rakenduse uuesti kompileerimine ja dialoogide salvestamine nõuavad operatsioonisüsteemi kirjutamise õiguseid, siis on soovitatav kogu antud tööga kaasapandud DVD sisu kopeerida kohalikku kataloogi.

DVD plaadil leiduvad:

- \Iaah – valmis kompileeritud rakendus lokaalseks kasutamiseks,
- \Iaah_src – rakenduse lähtekood,
- \Iaah_doc – uurimustöö kirjalikud failid. Bakalaureuse töö ja kasutajajuhend.

3.1 Keskkonna ülesseadmine

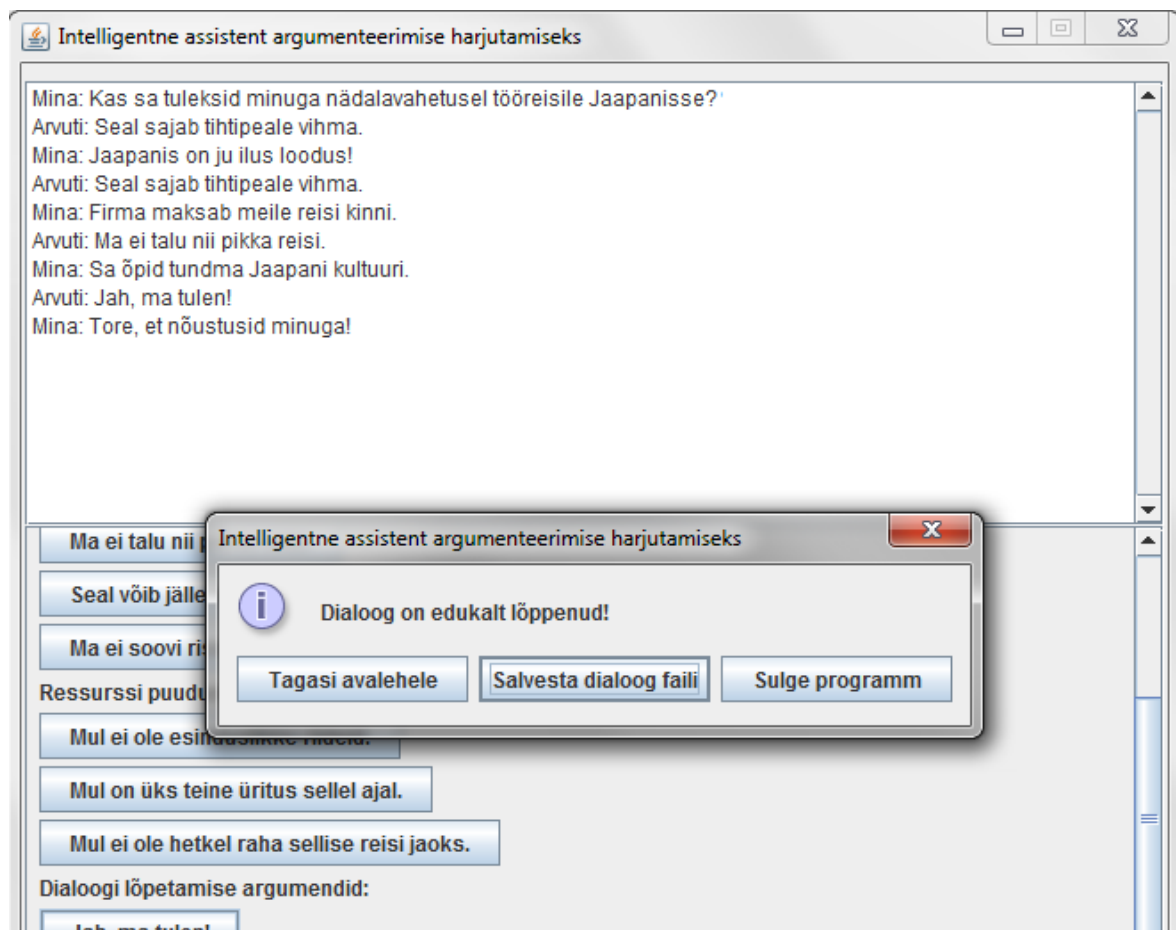
Selleks, et edukalt käivitada rakendus, on vaja teha järgmist.

1. Kopeerige DVD plaadi peal olevad kaustad ja failid kuskile kohalikku kataloogi.
2. Veenduge, et arvutisse on installeeritud Sun Microsystems Java Runtime Environment(JRE). Vastasel juhul saab viimase versiooni alla laadida aadressilt: <http://www.java.com/en/download/manual.jsp>.
3. Veenduge, et teie arvutis on seadistatud Java jaoks vajalikud keskkonna muutujad. Selleks kontrollige oma arvutis keskkonna muutujaid:
 - Windows XP:
 - a) Paremal klõps ikoonil „Minu arvuti“ ja valige atribuudid.
 - b) Avanenud aknast valige kaart „Täpsemalt“.
 - c) Klõpsake nupul „Keskkonna muutujad“.
 - d) Lisage kasutaja muutujate alla uus tee nimega „JAVA_HOME“ ja väärtuseks pange kataloogi asukoht, kus asub Java (näiteks C:\Program Files\Java\jdk1.6.0_21).
 - e) Lisage süsteemi muutuja „Tee“ (*Path*) alla samasugune kirje nagu punktis d), aga sama kirje lõppu lisage \bin. (näiteks C:\Program Files\Java\jdk1.6.0_21\bin).
 - Windows Vista/7:
 - a) Paremal klõps ikoonil „Minu arvuti“ ja valige atribuudid.
 - b) Avanenud aknast valige vasakult servast „Täpsemad süsteemi sätted“.
 - c) Avanenud aknast valige kaart „Täpsemalt“.
 - d) Klõpsake nupul „Keskkonna muutujad“.
 - e) Lisage kasutaja muutujate alla uus tee nimega „JAVA_HOME“ ja väärtuseks pange kataloogi asukoht, kus asub Java (näiteks C:\Program Files\Java\jdk1.6.0_21).
 - f) Lisage süsteemi muutuja „Tee“ (*Path*) alla samasugune kirje nagu punktis d), aga sama kirje lõppu lisage \bin. (näiteks C:\Program Files\Java\jdk1.6.0_21\bin).
 - Unixi bash shell:
 - a) Avage terminal ja kirjutage
export JAVA_HOME=<teie java kaust>
(näiteks export JAVA_HOME=/usr/java/jdk1.6.0_21)
 - b) Kirjutage terminali ka veel käsk
„export PATH=\$PATH:<Java asukoht arvutis>“
(näiteks export PATH=\$PATH:/usr/java/jdk1.6.0_21/bin)

4. Juhul, kui tahate ainult programmi käivitada, jätkake sammuga 6.
5. Lähtekoodi uuesti kompileerimise soovi korral tuleb kaust \Iaah_src avada mõne Java arenduskeskkonna abil (näiteks Eclipse, IntelliJ IDEA jms). Importida uus projekt kaustast ja kompileerida *mainframe.java*.
6. Liikuge kausta \Iaah ja tehke topeltklõps failil „Iaah.jar“. Kui punktis 3 sätestatud keskkonna muutujad on paigas, siis peaks rakendus ilma tõrgeteta avanema.

3.2 Dialoogi alustamine

1. Rakenduse esilehelt klõpsake nuppu „edasi“.
2. Järgnevalt lehelt valige teema, klõpsates ühe teema nime peale. Vaata ka Lisa 1.
3. Sama lehe alumiselt poolelt klõpsake dialoogi alustaja nupu peale. Vaata ka Lisa 1.
4. Dialoogi saab lauseid lisada, klõpsates vastava lause nupule.
5. Dialoog on lõppenud siis, kui klõpsate nupul „Jah, ma tulen!“ (olukorras, kus arvuti alustab) või kui olete arvuti suutnud ära veenda (olukorras, kus kasutaja alustab). Vaata joonis 3.



Joonis 3. Näide dialoogist, kus kasutaja veenab arvutit

3.3 Vestluse salvestamine

Illustreerivad joonised on toodud Lisas 3.

1. Dialoogi lõppedes kuvatakse kasutajale aken, kus on kolm valikut: tagasi avalehele, salvesta dialoog, sulge programm.
2. Klõpsake nupule „salvesta dialoog“.
3. Avaneb uus aken, millega saate valida asukoha dialoogi salvestamiseks.
4. Kirjutage lahtrisse „faili nimi“ (*file name*) soovitud nimi dialoogi logi-failile.
5. Klõpsake nupule „salvesta“ (*save*) ning teie dialoog salvestatakse teksti-faili formaadis teie poolt määratud asukohta.

4. Edasiarendamise võimalused

Uurimustöö teema juurde asudes anti autorile ette nõuded, mis pidid minimaalselt saama täidetud. Sealhulgas toodi välja, et on vaja luua rakendus, millega kasutaja saaks harjutada vaidlemist arvutiga erinevate argumentide kaudu. Samuti anti ette mõningaid kitsendusi, mis vähendasid töö mahtu ja seadsid piirid, et uurimustöö ei veniks liiga pikaks. Üheks suurimaks piiranguks osutus soovitus rakenduses mitte teha eesti keele lausete morfoloogilist, süntaktilist ja semantilist analüüsi. Ühest küljest vähendas see märgatavalt töö mahtu, teisest küljest raskendas loomuliku dialoogi tekkimist. Etteantud argumentidega on väga raske luua loomulikku vestlust inimese ja tehisintellekti vahel. Ülesande püstituses seati paika ka argumentide liigid, mida kasutada rakenduse loomisel. See kergendas algoritmi ehitamist, sest lihtsam on opereerida ette antud kindla arvu argumentidega, kuid samas need argumendid seadsid kitsendused dialoogi kulgemisele ning nagu näitas testimine, hakkasid vestlused arvuti ja inimese vahel toimuma kindla raami järgi.

Loodud rakendus saab olla aluspõhjaks arvutiga intelligentse argumenteerimise pidamiseks. Programmil on väga palju edasiarenduse võimalusi. Üheks peamiseks suunaks, võiks pidada eesti keele analüüsi sisse toomist. Nii saaks rakendusest eemaldada ette-defineeritud argumendid ja lubada kasutajal endal sisestada ükskõik milliseid lauseid. Abistamiseks saaks kasutusele võtta ka võtme-sõnad või -fraasid. Lisavõimalusena saaks aplikatsiooni viia eraldiseisvalt Java rakenduselt veebikeskkonda. HTML5 ja CSS-i koostööl oleks võimalik luua ilus ja atraktiivne kasutajaliides. Veebirakendusena töötades saaks luua andmebaasi, mis hoiaks endas kõiki toimunud dialooge. See aga looks võimaluse saata dialooge analüüsimiseks ja arhiveerimiseks Eesti Dialoogikorpusele [9]. Kõik see kokku muudaks programmi kasutajasõbralikumaks ja aitaks kaasa võimalikult loomulike vestluste toimumisele arvuti ja inimese vahel.

5. Mõned teised süsteemid

Bakalaureuse töö käigus valminud rakendus pole ainulaadne. Sarnaseid süsteeme leidub nii Eestis kui ka mujal maailmas. Selliste süsteemide, mis tegelevad inimeste vaheliste vestluste kirjaliku modelleerimisega, otstarve on lai. Enamus aplikatsioonidest on kirjutatud juturobotitena ehk rakendustena, mis vestlevad kasutajaga reaalselt.

Tänase päevani pole suudetud luua perfektset juturobotit, mis suudaks inimese panna uskuma, et ta räägib teise inimesega, mitte arvutiga. Seda probleemi käsitleb Turingi test [4], mille põhiülesandeks on modelleerida tehisintellekti, mis suudaks mõelda nagu inimene. Käesoleva ajani pole keegi suutnud kirjutada programmi, mis läbiks Turingi testi. Igal aastal toimub H.Loebneri auhinnavõistlus [5], mille peauhinna on sada tuhat dollarit ja kuldmedal, mis antakse inimesele, kes on suutnud luua programmi, mis läbiks Turingi testi. Siiani pole keegi veel kuldmedalit saanud. Igal aastal antakse auhinnavõistlusel parima programmi loojale kaks tuhat dollarit ja pronksmedal. Kuid sellised programmid tulevad toime vaid mõningate, mitte enamuse kohtunike äravõetamisega.

Üks esimesi programme, mis üritas inimesega pidada vestlust temale loomulikult lähedase keeles, loodi Joseph Weizenbaumi poolt 1966. aastal. Programm sai endale nime Eliza [6] ning talle anti isikukeskse psühhiaatri roll. Rakendus mängib nii-öelda psühhiaatrist tohtrit, kes kuulab inimest ja palub inimesel endal rääkida mingil teemal. Kõik küsimused, mida talle esitatakse, suunab ta tagasi küsijale. Näiteks kui Eliza käest küsitakse, mis on tema lemmik-film, siis Eliza küsib selle peale hoopis kasutaja käest tema lemmik-filmi, või hoopis küsib kasutaja käest, miks too tahab seda teada. Selline lähenemisviis sobib piiratud ainevaldkonnas ka keerukamate süsteemide loomiseks, kus on vaja rakendada dialoogi inimese ja arvuti vahel.

Eestis on ka loodud tehisintellekte, mis suudavad kasutajaga pidada vestlust, kuid enamuse on piiritletud mingi kitsa valdkonnaga. Näiteks Margus Treumuthi poolt loodud rakendus Kinoagent [7] annab kasutajale reaalselt informatsiooni hetkel kinos olevate filmide kohta. Aplikatsioon väldib teemaväliseid küsimusi ja kui kasutaja midagi ei ütle, siis agent ise tõstatab teema ning pakub kasutajale filme, mida võiks vaadata minna. Teise näitena võiks tuua J.Edenbergi bakalaureusetöö raames valminud Edelaraudtee tehisintellekti. Rakendus baseerub Eliza programmil, mille peale on ehitatud Edelaraudtee andmebaas. Mõlema näite puhul peetakse kasutajaga loomulikus eesti keeles vestlust skoobiga piiritletud valdkonnas.

Antud bakalaureusetöö käigus valminud rakendusel on nimetatud süsteemidega sarnaseid jooni. Põhiideeks on pidada vestlust kasutajaga loomulikus keeles. Teemad on piiritletud, mis tähendab, et suvalistel teemadel ei saa rääkida. Samas on ka erinevusi, näiteks käesoleva uurimustöö käigus valminud rakenduses ei saa kasutaja ise sisse trükkida lauseid, vaid peab opereerima ette antud lausetega.

6. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua rakendus, mille abil saaks kasutaja harjutada argumenteerimist tehisintellektiga. Vaadeldi selliseid dialooge, kus üks osaleja (A) püüab dialoogi käigus saavutada oma partneri (B) nõusoleku teha teatav tegevus D. Selleks uuriti esmalt erinevaid dialoogiteooriaid ja mudeleid ning eestikeelse dialoogi modelleerimise alal tehtud tööd. Lõpptulemusena valmis graafilise kasutajaliidese ja dialoogi salvestamise lisavõimalusega Java iseseisev rakendus, mis suudab kasutajaga pidada algtasemel dialoogi, kasutades etteantud argumente ette defineeritud teemal. Kasutaja võib endale võtta kas A või B rolli ja valida dialoogi teema (praegu kahe võimaliku hulgast). Mõlemad osalejad kasutavad etteantud argumente (eestikeelseid lauseid), mis on klassifitseeritud vastavalt sellele, missugust antud teemas vaatluse all oleva tegevuse aspekti nad rõhutavad (kasulikkus, kahjulikkus jne). Seega ei ole rakendatud eesti keele analüüsi. Selline piirang ei võimalda küll saada päris loomulikke eestikeelseid dialooge, kuid valminud programmiga suhtlemine peaks siiski aitama kasutajal harjutada erinevate argumentide valimist ja tagasilükkamist vaidluses teatud tegevuse tegemise üle.

7. Intelligent assistant for practicing argumentation

Summary

The objective of my bachelor thesis was to create a software which would help a user to practice argumentation with an artificial intelligent. To achieve this goal, dialogs, where one participator (A) tries to convince the other participator (B) to do certain activity D, were observed. Different dialog theories, models and already accomplished work in the field of dialog modeling in Estonian language were examined. The outcome of this process was an independent Java application, which could have a basic conversation with a user about predefined subjects using only predefined arguments, with graphical user interface and with an ability to save those dialogs. The user can either take role A or B. In addition the user can choose the topic of the conversation (at the moment 2 themes are available). Both participators must use given arguments (sentences in Estonian language) that are classified according to what part of the aspect of activity they emphasize. Therefore the analysis of Estonian language is not implemented. This limits the outcome of the natural dialogues in Estonian language, but the application should help a user to practice choosing different arguments and rejections in a conversation.

The structure of my Bachelor thesis can be divided into three different parts – base of the application, description of the application and analysis of the application. The first part gives an overview of the dialog theory. In these chapters different models are discussed and compared. In addition, important observations in modeling a dialog are presented. The second part describes the structure and design of the application. In these chapters an overview of used technologies, additional packages and user manual is given. In the third part, the possible uses of application are analyzed. In addition, similarities with similar already existing applications are pointed out. Also ideas for further development are given in these chapters.

In the appendixes, there are:

- screenshot of choosing the topic,
- screenshots of saving the dialog,
- example of the log file,
- two block-schemes describing the algorithm.

There's a DVD given with bachelor thesis, which contains:

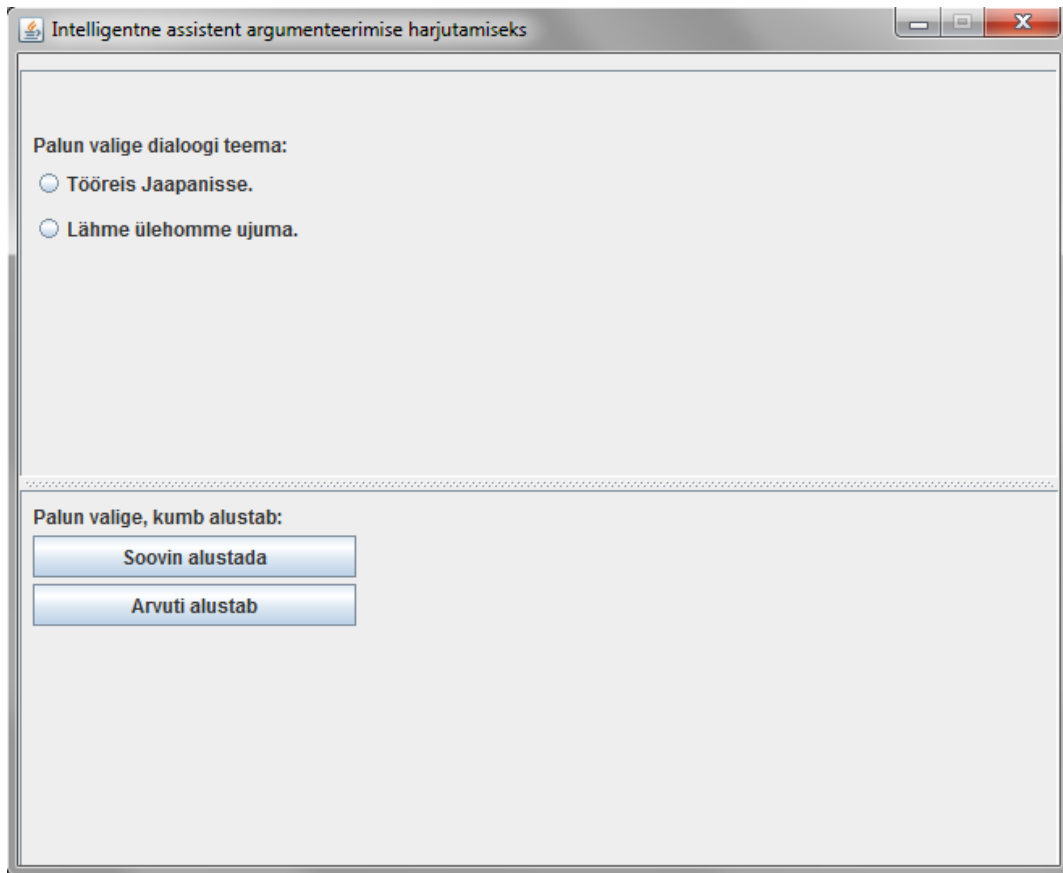
- source code of the software,
- necessary files for launching the application,
- user manual,
- written part of my bachelor thesis.

8. Kasutatud kirjandus

- [1] Mare Koit, Haldur Õim. Eestikeelse dialoogi modelleerimine, 2003.
<http://lepo.it.da.ut.ee/~koit/Dialoog/Artikkel/2003/DialoogKK1.rtf> . Vaadatud 1. juuni 2011.
- [2] Eesti keele seletav sõnaraamat.
<http://www.eki.ee/dict/ekss/index.cgi?Q=dialoog&F=M> . Vaadatud 1. juuni 2011.
- [3] Java lisateek MigLayout. <http://www.miglayout.com> . Vaadatud 1. juuni 2011.
- [4] *The Turing test page*. <http://www.fil.ion.ucl.ac.uk/~asaygin/tt/ttest.html> . Vaadatud 1. juuni 2011.
- [5] H.Loebneri auhind. <http://www.loebner.net/Prizetf/loebner-prize.html> . Vaadatud 1. juuni 2011.
- [6] C.Hayden. Eliza. <http://www.chayden.net/eliza/Eliza.html> . Vaadatud 1. juuni 2011.
- [7] M. Treumuth. Kinoagent. <http://www.dialoogid.ee/teatriagent> . Vaadatud 1. juuni 2011.
- [8] J.Edenberg. Edelarautee tehisintellekt. <http://www.joel.ee/eliza> . Vaadatud 1. juuni 2011.
- [9] Eesti Dialoogikorpus (EDiC). <http://www.cs.ut.ee/~koit/Dialoog/EDiC> . Vaadatud 1. juuni 2011.

9. Lisad

Lisa 1: Teema valimise leht (kaart kaks).



Intelligentne assistent argumenteerimise harjutamiseks

Palun valige dialoogi teema:

☐ Tööreis Jaapanisse.

☐ Lähme ülehommel ujuma.

Palun valige, kumb alustab:

Lisa 2: Väljavõtte dialoogi logifailist.

[2011-05-18 14:45:53.639] Arvuti: Kas sa tuleksid minuga nädalavahetusel tööreisile Jaapanisse?

[2011-05-18 14:45:54.168] Mina: Seal on praegusel aastaajal külm.

[2011-05-18 14:45:54.168] Arvuti: Jaapanis on ju ilus loodus!

[2011-05-18 14:45:54.479] Mina: Seal sajab tihtipeale vihma.

[2011-05-18 14:45:54.479] Arvuti: Jaapanis on lähed inimesed.

[2011-05-18 14:45:54.775] Mina: Toimunud katastroofi tõttu ei tahaks nagu eriti minna sinna.

[2011-05-18 14:45:54.775] Arvuti: Jaapanis on ju ilus loodus!

[2011-05-18 14:45:55.143] Mina: Ma ei talu nii pikka reisi.

[2011-05-18 14:45:55.143] Arvuti: Jaapanis on lähed inimesed.

[2011-05-18 14:45:55.447] Mina: Seal võib esineda jälle maavärinat ja tsunamit.

[2011-05-18 14:45:55.447] Arvuti: Firma maksab meile reisi kinni.

[2011-05-18 14:45:55.751] Mina: Ma ei soovi riskida tuumakiirituse saamisega.

[2011-05-18 14:45:55.751] Arvuti: Sa saad suurepärase kogemuse.

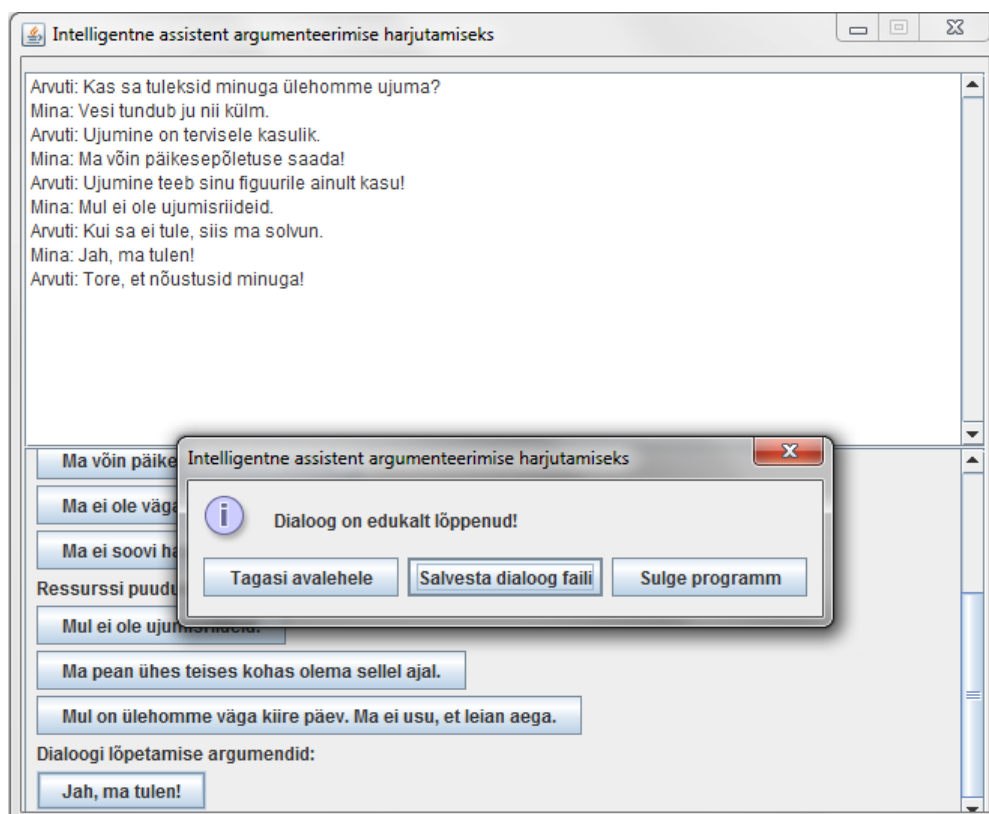
[2011-05-18 14:45:57.688] Mina: Mul ei ole esinduslikke riideid.

[2011-05-18 14:45:57.688] Arvuti: Kui sa ei tule, siis ülemus võib sind vallandada.

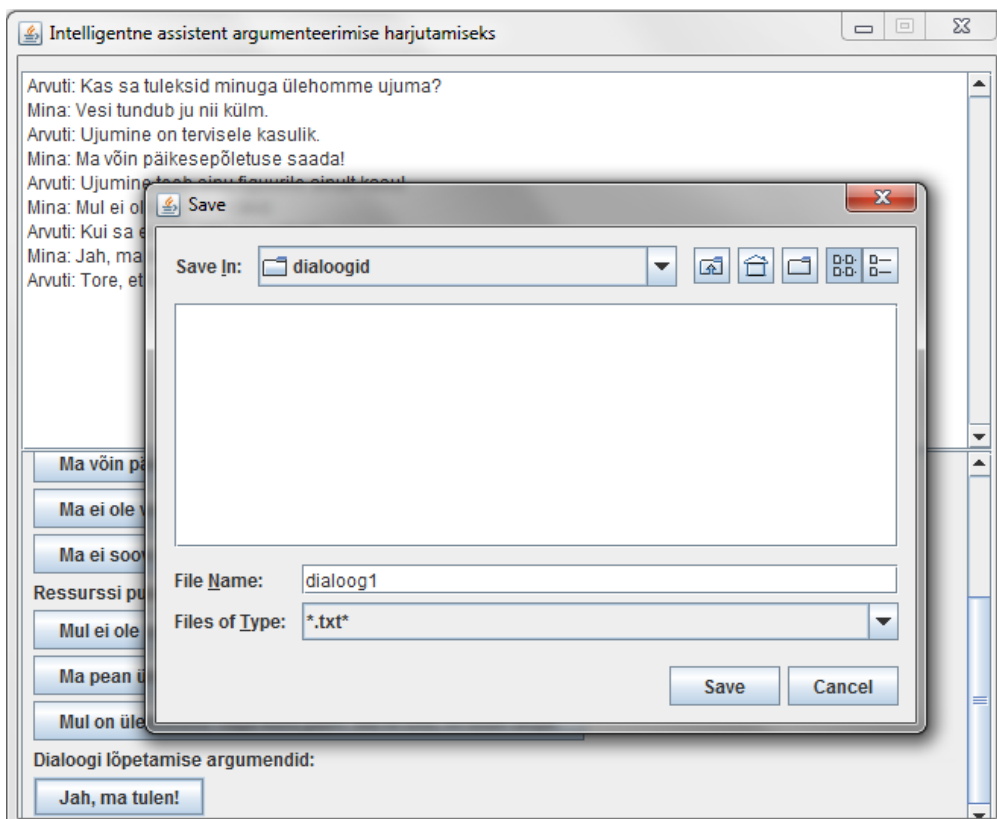
[2011-05-18 14:45:59.712] Mina: Jah, ma tulen!

[2011-05-18 14:45:59.712] Arvuti: Tore, et nõustusid minuga!

Lisa 3. Dialoogi salvestamise ekraanipilt

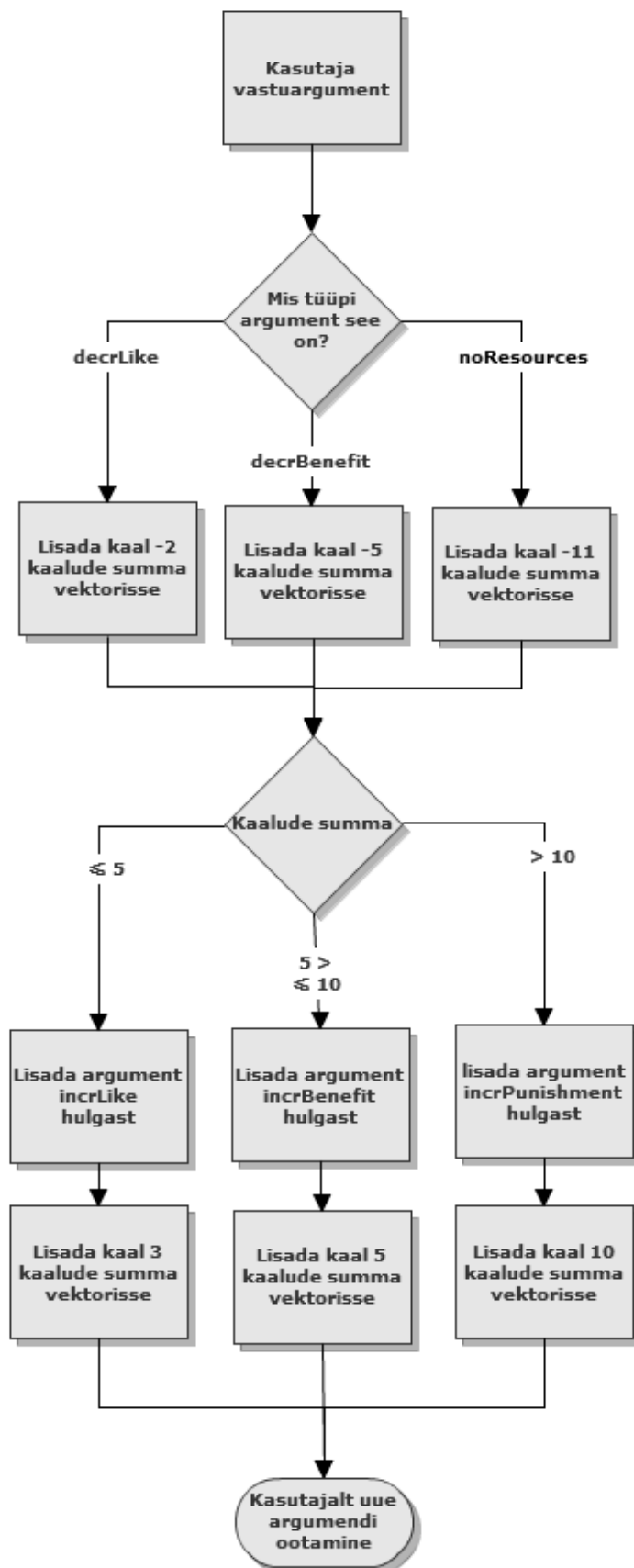


Joonis 4. Dialoogi lõpp

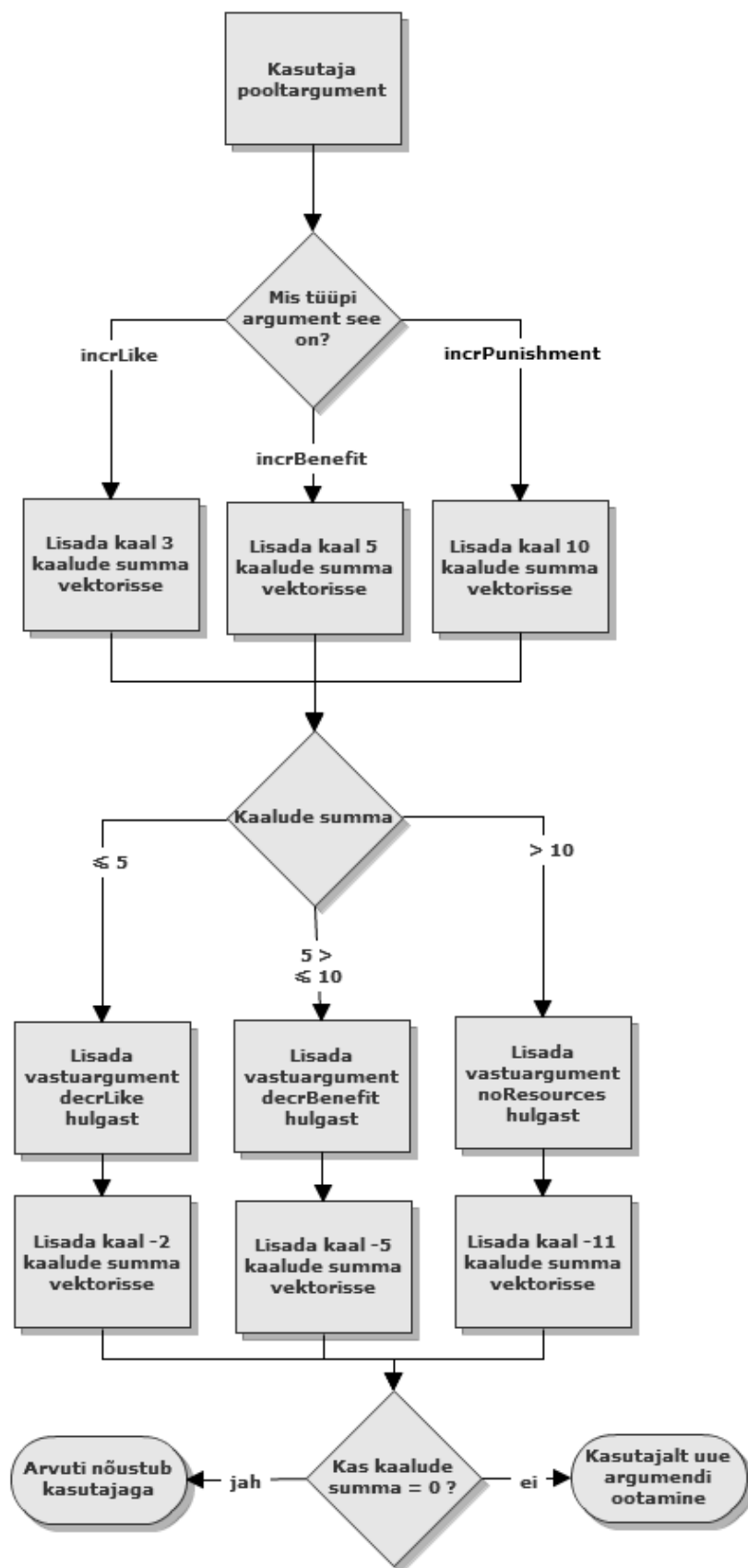


Joonis 5. Dialoogi salvestamine

Lisa 4. Algoritmi plokk-skeemid



Joonis 6. Algoritmi töö, kui kasutaja on B rollis



Joonis 7. Algoritm töö, kui kasutaja on A rollis